

**Document some specific cases in which the Levenshtein-distance spelling corrector works well and in which it works poorly. Why is it working poorly and how could it be improved?**

1. Our original spelling corrector fails to correct some wrong words like “worng” and “htps” because they are improperly spelled in the given word list as input. To fix this problem, we add a new function “corrector.py” using the weighted combination. The weighting factor  $p$  we use in our program is 0.1.
2. Sometimes if we accidentally reverse two letters in a word, using Levenshtein-distance method will usually not fix it to the correct word, but correct it to a word that is less distant from it, because the letter-reordered words tend to have more distance. For example, if we wrongly type “worng” when we use “wrong”, it will return “worn” instead of “wrong”.
  - A way to fix it is finding other methods rather than Levenshtein-distance to correct words.
3. Because different words have different using frequency, if they have the same distance to the wrong word, the frequency will dominate the corrector. For instance, if we want to type “apple” but wrongly type “aple”, the corrector will find “able” instead of “apple” because the frequency of using “able” is higher than “apple”.
  - A way to fix it is finding other methods rather than Levenshtein-distance to correct words.
  - The other way is analyzing phrase to correct the words instead of using just one word.